In the
United States Court of Appeals
For the Seventh Circuit

No. 01-4222

NICHOLAS E. EUSTACE, *et al.*,

Petitioners-Appellants,

v.

COMMISSIONER OF INTERNAL REVENUE,

Respondent-Appellee.

Appeal from the United States District Court Tax Court
Nos. 21088-96 *et al.*—**Maurice B. Foley**, *Judge.*

ARGUED NOVEMBER 7, 2002—DECIDED DECEMBER 13, 2002

Before BAUER, EASTERBROOK, and MANION, *Circuit Judges.*

EASTERBROOK, *Circuit Judge.* Applied Systems, a Subchapter S corporation, develops and sells software that independent insurance agencies use to manage their businesses. During the early 1990s Applied Systems improved its software package, and its investors (the taxpayers in this case) want to take a tax credit under 26 U.S.C. §41 based on the amount by which Applied Systems increased its R&D expenses during these years. Taxpayers who have sought a credit under §41 for commercial software development have been uniformly unsuccessful. See, e.g., *Tax & Accounting Software Corp. v. United States*, 301 F.3d 1254 (10th Cir. 2002); *Wicor, Inc. v. United States*, 263 F.3d

659 (7th Cir. 2001); *United Stationers, Inc. v. United States*, 163 F.3d 440 (7th Cir. 1998); *Norwest Corp. v. CIR*, 110 T.C. 454 (1998). The only exception—on which the taxpayers in this case principally rely—is the district court's decision in *Tax & Accounting*, 111 F. Supp. 2d 1153 (N.D. Okla. 2000), which the tenth circuit reversed after they filed their brief. Applied Systems' claim is no better than the others.

The evidence at trial in the Tax Court shows that Applied Systems engaged in normal software development. During the early 1990s it enhanced its software package so that it could handle additional ratings computations, so that it could handle transactions between insurers and agencies, so that multiple people could work on the same customer file simultaneously without corrupting or overwriting each others' changes, and so that more functions could be handled in a given amount of random access memory. It discarded a word processing module licensed from another vendor, replacing it with a simple text editor with reduced memory demands but good form-letter-generation features. These are not the only changes, but they show the character of the work. None of it was pioneering; all of it entailed variations on themes long used by other developers. The Commissioner conceded that this work met many of the requirements in §41 but contested several others, of which only two require attention: that the research be "undertaken for the purpose of discovering information which is technological in nature" (§41(d)(1)(B)(i)), and that the activities "constitute elements of a process of experimentation" (§41(d)(1)(C)). The Tax Court concluded that Applied Systems flunked both tests—the former because it did not produce an "innovation in underlying principle" and the latter because the research in question was not designed to dispel uncertainty about the technological possibility of developing software of this kind. T.C. Memo 2001-66 (2001).

Taxpayers concede that if §41 means what the Tax Court thought, then they lose. Applied Systems has not tried to show that its software embodies any leap in information technology or that there was any doubt about the technological ability to produce software of this kind. But it contends that §41 does not set so high a standard and that industrious development of software through a process of trial and error meets the statutory standard. The difficulty with this position is that the Tax Court did not get its legal views out of thin air. This court announced them in *United Stationers*, and *Wicor* declined an opportunity to revisit the subject. *United Stationers* held that software development satisfies the statutory technological-information requirement only if "the research is intended to expand or refine existing principles of computer science" and the resulting information is "of broad effect" (163 F.3d at 444). As for experimentation, we held that the taxpayer must formulate and test hypotheses in order to dissipate uncertainty about the possibility of success, a standard that fine-tuning (or debugging) of computer programs does not satisfy. *Id.* at 445-46.

Although the word "experiment" has many shadings in common speech, we held that as used in §41 it has the *scientific* sense of forming and testing hypotheses rather than the lay (or even engineering) sense of trial and error. Galileo engaged in experiments about acceleration when he rolled balls down an inclined plane. An auto manufacturer trying different nozzles from those on hand to find the one that applies the smoothest coat of paint is not engaged in "experimentation" under this view, nor is a software developer trying different methods to implement a feature accompanied by maximum execution speed and minimum demand on system resources such as RAM. Tinkering differs from experimentation in the vocabulary of research—and §41 is about research, and thus about use of the scientific method. Authors and movie makers playing with sentences and scenes to find what most impresses the public are not doing scientific research using "experimentation". Just so with software. Developers are authors too; that they write lines of code readable by machines rather than lines of words readable by people does not fundamentally change the nature of the task and make one form of writing "experimentation" when the other is not. Experimentation is a subset of all steps taken to resolve uncertainty; otherwise searching for a place to park a car would be a "process of experimentation".

The tenth circuit has disagreed with our understanding of §41. It held in *Tax & Accounting Software* that the technological-information requirement can be satisfied by new knowledge that is less a step forward than *United Stationers* required—but that "information must be separate from the product that is actually developed." 301 F.3d at 1262. With respect to experimentation, the tenth circuit was not receptive to the idea that only hypothesis formulation and testing fits, but it still held that the taxpayer must establish that testing was designed to overcome uncertainty about whether the desired end result was technologically feasible. *Id.* at 1264-68. Applied Systems cannot meet these definitions any more than those in *United Stationers*.

In the long run, neither our view nor the tenth circuit's has staying power. Both *United Stationers* and *Tax & Accounting Software* analyzed §41 without the benefit of the regulations that are supposed to illuminate the path to decision. Section 41's predecessor was enacted in 1981, and §41 has been on the books in its current form since 1986, but the Internal Revenue Service has yet to promulgate the regulations that are important to this statutory design. (Section 41 refers ten times to regulations that the Secretary of the Treasury is to develop and issue.) The most recent draft was published almost a year ago, 66 Fed. Reg. 66,362 (Dec. 26, 2001), and has not been made final.

Applied Systems asks us to discard the approach of *United Stationers* and use the one found in the draft regulations. That would not be sound, for two reasons: first, proposed regulations have no legal effect; second, the draft says that when final the regulations will apply only to taxable years ending on or after December 21, 2001. So Applied Systems gets no solace from this source even if the regulatory approach would favor its position (which we doubt, given that the regulations essentially track *United Stationers*' definition of "experimentation" as use of the scientific method).

One other line of argument requires only brief mention. Applied Systems thinks that we should disregard *Wicor* and *United Stationers* on the ground that the credit sought by those taxpayers was covered by §41(d)(4)(E), which disqualifies the costs of internal-use software except to the extent allowed by (nonexistent) regulations. In both *Wicor* and *United Stationers* the taxpayer contracted with a consulting firm to develop software that the taxpayer would use in its own business, while Applied Systems developed software for sale to customers. Yet §41(d)(4)(E) has nothing to do with the definitions in §41(d)(1), and *Wicor* stopped there. *United Stationers* held that the taxpayer lost under both §41(d)(1) and §41(d)(4)(E). That an opinion contains multiple grounds of decision does not justify disregarding any of them; it would be no more sound to throw out *United Stationers*' interpretation of §41(d)(1) than to disregard its interpretation of §41(d)(4)(E) (which was that contracting-out the development of software for in-house use fits within the §41(d)(4)(E) disqualification). Sections 41(d)(1) and (d)(4) are independent rules, which deserve, and have received, independent constructions.

Whether we apply the definitions articulated in *United Stationers* or the competing interpretation from *Tax & Accounting Software*, simple industrious software development does not qualify for the §41 tax credit. Accordingly, the judgment of the Tax Court is

AFFIRMED.

A true Copy:

Teste:

Clerk of the United States Court of
Appeals for the Seventh Circuit